CVCoders at SemEval-2024 Task 4: Unified Multimodal Modelling For Multilingual Propaganda Detection in Memes

Fatemezahra Bakhshande and Mahdieh Naderi and Sauleh Etemadi
Iran University of Science and Technology
{bakhshande.ghazal, mahdieh9816, sauleh}@gmail.com

Abstract

This paper presents our approach to the SemEval 2024 Task 4 on "Multilingual Detection of Persuasion Techniques in Memes." We address the challenge of identifying persuasion techniques in textual and multimodal meme content using a combination of preprocessing techniques and Uni-modal models. Leveraging advanced preprocessing methods, including the OpenAI API for text data, we achieved improved data quality. Our model architecture combines VGG for image feature extraction and GPT-2 for text feature extraction, yielding superior performance. To mitigate class imbalance, we employed Focal Loss as the loss function and AdamW as the optimizer. Experimental results demonstrate the effectiveness of our approach, achieving competitive performance in the task.

1 Introduction

The SemEval 2024 Task 4 [1]focuses on the multilingual detection of persuasion techniques in memes, a crucial endeavor in combating disinformation campaigns prevalent on social media platforms. Memes, being potent vehicles for influencing public opinion, necessitate robust methods for identifying rhetorical and psychological techniques embedded within their textual and visual content. This task spans multiple languages, including Bulgarian, English, and North Macedonian, underscoring the global significance of addressing online misinformation (Dimitrov et al., 2024).

Our system employs a combination of pre-trained models for text and image processing to tackle the challenge posed by Subtask 2b of Task 4. Specifically, we utilize pre-trained language models such as XLM-RoBERTa and GPT-2 for textual feature extraction, while employing VGG and ViT models for image feature extraction. This multimodal approach allows us to effectively capture both textual and visual cues present in memes.

Through our participation in this task, we discovered the importance of advanced preprocessing techniques, particularly in cleaning and standardizing textual data extracted from memes. Leveraging the GPT API for text preprocessing and NLTK for further cleaning proved instrumental in enhancing the quality of our training data. Additionally, we observed the significance of model selection and hyperparameter tuning in achieving competitive performance. Despite encountering challenges in cleaning textual data, our system achieved promising results, demonstrating the efficacy of our approach.

2 Background

The task at hand, Subtask 2b of SemEval-2024 Task 4, revolves around the multilingual detection of persuasion techniques in memes. Memes, which are widely circulated across social media platforms, often contain subtle rhetorical and psychological strategies aimed at influencing public opinion. The goal of the task is to develop models capable of identifying these persuasion techniques embedded within the textual and visual content of memes.

The input to the task consists of textual and visual data extracted from memes in various languages, including Bulgarian, English, and North Macedonian. The textual content of memes may contain linguistic elements such as catchphrases, slogans, or captions, while the visual component typically comprises images or graphics. For example, a meme may feature a humorous image accompanied by a caption containing persuasive language or propaganda.

As for our participation, we focused on Subtask 2b of Task 4, which involves analyzing the presence of persuasion techniques in memes using both

[1] https://propaganda.math.unipd.it/semeval2024task4

Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024), pages 1912–1918
June 20-21, 2024 ©2024 Association for Computational Linguistics

textual and visual features. Our approach combines advanced preprocessing techniques with state-of-the-art models to effectively tackle this challenging task. We draw inspiration from related work in the fields of natural language processing and computer vision, leveraging pre-trained models and techniques to enhance the accuracy and efficiency of our system.

2.1 Related Work

Generative Pre-trained Transformer 2 (GPT-2) is a large language model developed by OpenAI, pre-trained on a dataset of 8 million web pages. It exhibits general-purpose learning capabilities, enabling various tasks such as text translation, question answering, summarization, and text generation (Vincent, 2019; OpenAI, 2019; Piper, 2019).

XLM-R, a large-scale multilingual language model, demonstrates significant performance gains across diverse cross-lingual tasks, outperforming mBERT on tasks such as XNLI and MLQA (Conneau et al., 2020).

Researchers have proposed modified VGG-16 architectures for datasets like CIFAR-10, achieving improved performance with stronger regularization techniques and Batch Normalization (Liu and Deng, 2015).

Vision Transformer (ViT) demonstrates the effectiveness of pure transformer architectures applied directly to image patches for image classification tasks, achieving excellent results compared to convolutional networks (Dosovitskiy et al., 2020).

In recent years, SemEval has incorporated memes into some of its projects, such as Task 6 in 2021[2].

SemEval-2021 Task 6 focused on detecting persuasion techniques in memes, attracting significant participation and highlighting the importance of modeling interactions between text and image modalities (Dimitrov et al., 2021).

SemEval-2023 Task 3[3]. addressed persuasion techniques detection with a multilingual dataset, achieving competitive results using a fine-tuned XLM-RoBERTa large model (Hromadka et al., 2023).

3 System overview

Our system for Subtask 2b of Task 4 in SemEval 2024 employs a combination of algorithms and modeling decisions to detect persuasion techniques in memes based on both textual and visual content. In this section, we outline the key components of our system, including preprocessing steps, model architectures, and training procedures.

3.1 Text Preprocessing

The textual content extracted from memes often contains noise and irrelevant information, which can adversely affect the performance of downstream tasks such as persuasion technique detection. To address these challenges, we employ a series of preprocessing steps to clean and standardize the text data.

3.1.1 OpenAI API for Initial Preprocessing

We utilize the OpenAI API for initial text preprocessing, leveraging its advanced natural language processing capabilities to handle common challenges encountered in meme text extraction. The API effectively identifies and removes extraneous information such as dates, usernames, and additional text that may accompany the original meme content. By leveraging the power of the OpenAI API, we ensure that the text data fed into our system is clean and devoid of irrelevant noise.

3.1.2 Further Cleaning with NLTK

Following the initial preprocessing step, we employ the Natural Language Toolkit (NLTK) for further cleaning and normalization of the text data. The NLTK library provides a wide range of text processing tools, including tokenization, stemming, and stop-word removal, which help standardize the textual content extracted from memes.

3.1.3 Manual Data Correction

Initially, we trained the data without text and solely using images, achieving 45% F1 macro on dev set. So, the major challenge we faced was how to incorporate text.

In instances where the automated text extraction process yielded inaccurate results, manual intervention was necessary to correct these errors. This phase involved a meticulous review of the textual content of problematic memes by human annotators, who then made the necessary adjustments to rectify any discrepancies.

[2] https://propaganda.math.unipd.it/semeval2021task6/

[3] https://propaganda.math.unipd.it/semeval2023task3/

This manual correction process was crucial for ensuring the accuracy and reliability of the textual data used in our system. By meticulously aligning the extracted text with the actual content depicted in the meme images, we mitigated potential biases and inconsistencies that could adversely affect the performance of our system.

In Listing 1, it can be observed that the textual content provided in the 'text' field ("@:\nDer") differs from the text contained within the associated image ('prop_meme_4499.png').

```
{
    "id": "25064",
    "text": "@:\\nDer",
    "image": "prop_meme_4499.png",
    "label": "propagandistic"
}
```

Listing 1: Sample Data Illustrating Textual Content Discrepancy in Manual Data Correction

The actual text present in the meme image is as follows: "Donald Trump Jr. @DonaldTrumpJr.8s\nMuppets have races now? So based on the orange\nI'm guessing Ernie is a Trump and must be \ncancelled immediately!!!\n\nABC News @ABC.7h\nAt only 7 years old, Ji-Young is making history\nas the first Asian American muppet in the\n"Sesame Street" canon. abcn.ws/3FyppJx\n\n'SESAME STREET' DEBUTS\nASIAN AMERICAN MUPPET\nabc NEWS\n\nAP PHOTO/NOREEN NASIR"

3.2 Image Preprocessing

Image preprocessing involves standard techniques such as resizing and normalization to enhance the quality and diversity of the image data. We employ the PyTorch framework for image preprocessing, utilizing built-in functions for resizing and normalization.

In Figure 1, we illustrate the data structure and preprocessing steps employed in our approach.

3.3 Feature Extraction

For textual feature extraction, we fine-tune pre-trained language models, including XLM-RoBERTa and GPT-2, on the meme text data. These models capture semantic and syntactic information embedded in the textual content, enabling effective representation learning for downstream tasks. For image feature extraction, we explore both convolutional neural networks (CNNs) such as VGG and vision transformers (ViTs) to extract visual features from memes. The extracted features from both modalities are concatenated to form a multimodal feature representation of the memes.

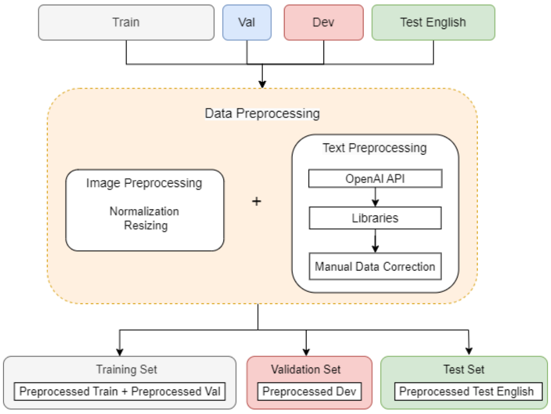

Figure 1: Diagram illustrating the Data structure and preprocessing steps

3.4 Model Architecture

Our model architecture consists of a multimodal fusion layer followed by a classification layer. The multimodal fusion layer combines textual and visual features using concatenation to integrate information from both modalities. The classification layer employs a binary classification approach to predict the presence or absence of persuasion techniques in memes.

In Figure 2, we present the architecture of our Best model, which combines VGG-16 and GPT-2 for Subtask2b.

3.5 Multilingual Considerations

Given the multilingual nature of the task, one initial consideration was how to effectively handle language diversity within the dataset. Initially, we contemplated translating the data into English to leverage state-of-the-art monolingual language models such as BERT. However, inspired by insights from the top-performing submission in last year's TASK 3, (Hromadka et al., 2023) we recognized the efficacy of utilizing pre-trained multilingual models like GPT and XLM-RoBERTa.(Liu et al., 2019) This approach proved advantageous, allowing our system to effectively analyze memes across different languages without the need for explicit translation.

3.6 Overfitting Mitigation Strategies

During the development phase, we encountered challenges related to model overfitting, particularly when using complex architectures such as

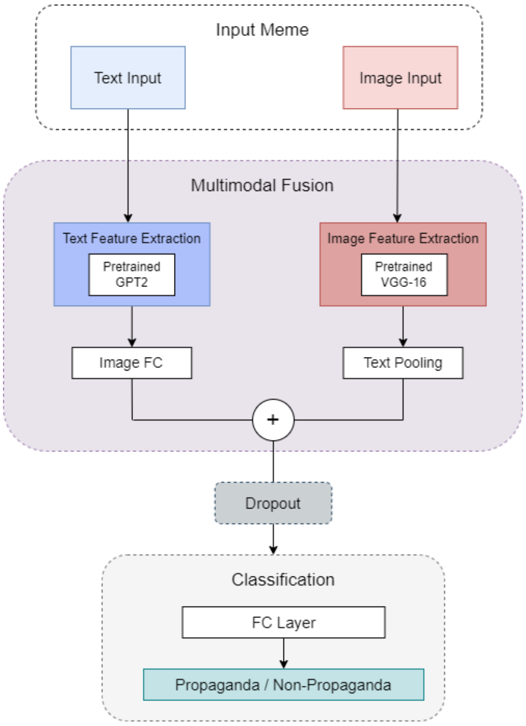

Figure 2: Model architecture combining VGG-16 and GPT-2 for Subtask2b.

a combination of XLM-RoBERTa for text processing and VGG for image analysis. Without proper normalization, our initial model exhibited signs of overfitting, compromising its generalization capabilities. To address this issue, we implemented regularization techniques, including dropout layers, to prevent overfitting and enhance the robustness of our model. These measures proved instrumental in stabilizing the training process and improving the overall performance of our system.

3.7 Training Procedure

We train our model using a combination of supervised learning and fine-tuning techniques. We train the model with the training data merged with the validation data. We employ Focal Loss as the loss function to address class imbalance and AdamW optimizer for gradient descent optimization. Hyperparameters such as learning rate, batch size, and dropout rate are tuned using grid search and cross-validation on the dev set.

3.8 Evaluation and Results

The performance of our system is evaluated using standard evaluation metrics such as macro-F1 score on the test set. We compare our results with baseline models to assess the effectiveness of our approach.

3.9 System Variants

We explore multiple system configurations, including variations in model architectures, preprocessing techniques, and hyperparameter settings. Each variant is evaluated and compared based on its performance on the validation set, allowing us to identify the most effective configuration for the task.

4 Experiment Setup

In this section, we detail the experimental setup used to train and evaluate our system for Subtask 2b of TASK4 2024. This task is a multi-model binary classification task.

4.1 Data Splitting

As shown in Table 1, in this task, we have 1200 samples in the train dataset, 150 samples in the validation dataset, and 300 samples in the `dev_unlabeled` dataset. Initially, the labels for the development dataset were unavailable to be used for testing purposes. However, eventually, these labels were fully accessible under the name of `dev_gold_labels` to the participants, and a dataset consisting of 600 samples was curated to serve as the test dataset, for which the labels have not yet been released.

Data Set/Label	Propagandistic	Non-Propagandistic
Train	800	400
Validation	100	50
Development	200	100

Table 1: Distribution of Datasets

At the beginning of our work, we utilized the same provided training dataset to train our initial model. However, we noticed that the model's accuracy on the training data reached 90% after 2 epochs, but the accuracy on the validation data was not as promising. Despite adjusting hyperparameters, this discrepancy in accuracy did not improve. Therefore, we decided to proceed by using the entire dataset for training our models. This expanded dataset significantly improved the model's performance on the test data.

4.2 Loss Function

After examining the labeled training and validation datasets, we noticed that the data distribution across classes is not uniform. Therefore, we opted to use focal loss alongside binary cross-entropy as the loss function.(Terven et al., 2023)

The formula for Binary Cross-Entropy (BCE) is given by:

$$BCE = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\hat{y}_i) + (1 - y_i)\log(1 - \hat{y}_i)] \quad (1)$$

Where:

N is the number of samples,

y_i is the true label of sample i,

$\hat{y}_i$ is the predicted probability of sample i.

The formula for Focal Loss combined with Binary Cross-Entropy (BCE) is given by:

$$\begin{aligned} FocalLoss + BCE = -\frac{1}{N}\sum_{i=1}^{N}[(1-\hat{y}_i)^{\gamma} y_i \log(\hat{y}_i) \\ + (1-y_i)^{\alpha} \hat{y}_i \log(1-\hat{y}_i)] \end{aligned} \quad (2)$$

Where:

N is the number of samples,

y_i is the true label of sample i,

$\hat{y}_i$ is the predicted probability of sample i,

α and γ are hyperparameters controlling the balance and focusing strength

4.3 Hyperparameter Tuning

Hyperparameter tuning played a crucial role in optimizing model performance. We experimented with various hyperparameters, including learning rates, batch sizes, and thresholds, to find the optimal configuration.

Further training parameters are specified in Table 2, in addition to those mentioned above.

Params	Value
number of train epoch	10
train batch size	32
validation batch size	32
weight decay	0.001
learning rate	$1e^{-3}$
threshold	0.39

Table 2: training hyperparameters

Moreover, In Figure 3, the impact of thresholds on Precision, Recall, F-score, and F1-macro metrics is visualized. This analysis provides insights into the trade-offs between these metrics, guiding the selection of an optimal threshold for model evaluation and decision-making.

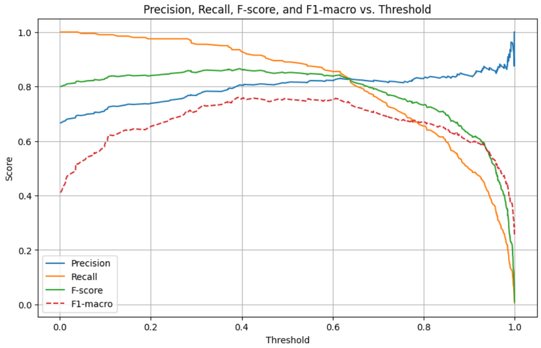

Figure 3: Precision, Recall, F-score, and F1-macro vs. Threshold on the Development Set

4.4 Tools and Libraries

Our system leveraged several external tools and libraries, including:

- OpenAI GPT API[4] for text preprocessing
- PyTorch[5] deep learning framework (v1.9.0) for model implementation
- Hugging Face Transformers[6] library (v4.11.3) for accessing pre-trained language models

4.5 Evaluation Measures

The evaluation of our system's performance was based on macro-F1 score, which accounts for precision and recall across all classes. This metric provides a comprehensive assessment of the model's ability to detect persuasion techniques in memes, considering both true positive and false positive rates (Powers, 2007).

The precision, recall, and F1 score are calculated as follows :

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

$$F\_1Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (5)$$

Where:

TP is the total number of true positives,

FP is the total number of false positives,

FN is the total number of false negatives.

[4] https://openai.com/gpt

[5] https://pytorch.org

[6] https://huggingface.co/transformers

F1-macro is calculated as the average of the F1 scores for each class in the classification. It gives equal weight to each class, regardless of its size. The formula for F1-macro is:

$$F1_{\text{macro}} = \frac{1}{N} \sum_{i=1}^{N} F1_i \tag{6}$$

Where:

N is the number of classes,
$F1_i$ is the F1 score for class i.

F1-micro is calculated by considering the total number of true positives, false negatives, and false positives across all classes. It gives equal weight to each instance, regardless of its class. The formula for F1-micro is:

$$F1_{\text{micro}} = \frac{2 \times TP}{2 \times TP + FP + FN} \tag{7}$$

Both F1-macro and F1-micro are commonly used to evaluate the performance of classification models, especially in situations where class imbalance exists.(Opitz and Burst, 2019)

5 Results

5.1 Main Quantitative Findings

Our system achieved moderate performance in Subtask 2b of TASK4 2024. On the English test dataset, it attained an F1 macro score of 0.67 and an F1 micro score of 0.74.

The evaluation results of four different model combinations, utilizing the best possible threshold based on the F1 macro on the English Dev dataset, are presented in Table 3. These combinations include VGG + XLM-RoBERTa, VGG + GPT-2, ViT + XLM-RoBERTa, and ViT + GPT-2.

Model	F1 macro	F1 macro Best Threshold
VGG + XLM-RoBERTa	0.58	0.63
VGG + gpt-2	0.71	0.76
ViT + XLM-RoBERTa	0.40	0.53
ViT + gpt-2	0.35	0.51

Table 3: Dev Set Result

Furthermore, the best model, GPT + VGG, was tested on the test data across three languages, and its results are shown in Table 4.

5.2 Quantitative Analysis

To gain deeper insights into our system's performance, we conducted ablation studies and compared different design decisions to identify optimal configurations. We utilized the entire training dataset for these analyses, employing a combination of train, validation, and gold_unlabeled data for training and validation purposes.

Through systematic experimentation, we observed that incorporating focal loss with sigmoid binary activation significantly improved the model's performance, particularly in handling class imbalance issues. Furthermore, training the model using gold_unlabeled data as an additional validation set resulted in notable enhancements in accuracy.

6 Conclusion

This paper presents our approach to the SemEval 2024 Task 4 on "Multilingual Detection of Persuasion Techniques in Memes." Leveraging preprocessing techniques and a multimodal model architecture combining VGG for image features and GPT-2 for text features, our system achieved competitive results on the test dataset in Subtask 2b.

Furthermore, our findings suggest that GPT-2 exhibits greater generalizability than XLM-RoBERTa, with a lower limit on the number of tokens. Insights from our experiments underscore the potential of pre-training models on similar data to enhance performance and generalization.

Looking ahead, future work could focus on pre-training models on meme-specific data and refining preprocessing techniques for extracted text. The large amount of available data presents an opportunity to delve deeper into this aspect, potentially improving model accuracy and robustness in meme analysis tasks.

References

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. arXiv:1911.02116v2.

Dimitar Dimitrov, Firoj Alam, Maram Hasanain, Abul Hasnat, Fabrizio Silvestri, Preslav Nakov, and Giovanni Da San Martino. 2024. Semeval-2024 task 4: Multilingual detection of persuasion techniques in memes. In *Proceedings of the 18th International Workshop on Semantic Evaluation*, SemEval 2024, Mexico City, Mexico.

Language	F1 macro	Baseline F1 macro	F1 micro	Baseline F1 micro
English	**0.67398**	0.25000	**0.74000**	0.33333
Bulgarian	**0.51637**	0.16667	**0.74000**	0.20000
North Macedonian	**0.57653**	0.09091	**0.79000**	0.10000

Table 4: Best Model Result On Test Set

Dimitar Dimitrov, Bishr Bin Ali, Shaden Shaar, Firoj Alam, Fabrizio Silvestri, Hamed Firooz, Preslav Nakov, and Giovanni Da San Martino. 2021. Semeval-2021 task 6: Detection of persuasion techniques in texts and images. *Proceedings of the thirteenth Workshop on Semantic Evaluation*, Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2021):70–98.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2020. Unsupervised cross-lingual representation learning at scale.

Timo Hromadka, Timotej Smolen, Tomas Remis, Branislav Pecher, and Ivan Srba. 2023. KInITVeraAI at SemEval-2023 task 3: Simple yet powerful multilingual fine-tuning for persuasion techniques detection. *Proceedings of the Seventeenth Workshop on Semantic Evaluation*.

Shuying Liu and Weihong Deng. 2015. Very deep convolutional neural network based image classification using small training sample size. In *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 1–5, Kuala Lumpur, Malaysia. IEEE.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

OpenAI. 2019. Gpt-2: 1.5b release.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*. Submitted on 8 Nov 2019 (v1), last revised 8 Feb 2021 (this version, v3).

Kelsey Piper. 2019. A poetry-writing ai has just been unveiled. it's ... pretty good. *Vox*.

David M W Powers. 2007. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. *Journal of Machine Learning Technologies*, page 37–63.

Juan Terven, Diana M. Cordova-Esparza, and Alfonso Ramirez-Pedraza. 2023. Loss functions and metrics in deep learning. *Journal of Machine Learning Research*.

James Vincent. 2019. Openai has published the text-generating ai it said was too dangerous to share. *The Verge*.